BUILDING 3D-STRUCTURES WITH AN INTELLIGENT ROBOT SWARM

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Yiwen Hua

May 2018

ABSTRACT

This research is an extension to the TERMES system, a decentralized autonomous construction team composed of swarm robots building 2.5D structures[1], with custom-designed bricks. The work in this thesis concerns 1) improved mechanical design of the robots, 2) addition of heterogeneous building material, and 3) an extended algorithmic framework to use this material. In order to lower system cost and maintenance, the TERMES robot is redesigned for manufacturing in low-end 3D printers and the new drive train, including motor adapters and pulleys, is based on 3D printed components instead of machined aluminum. The work further extends the original system by enabling construction of 3D structures without added hardware complexity in the robots. To do this, we introduce a reusable, spring-loaded expandable brick which can be easily manufactured through one-step casting and which complies with the original robots and bricks. This thesis also introduces a decentralized construction algorithm that permits an arbitrary number of robots to build overhangs over convex cavities. To enable timely completion of large-scale structures, we also introduce a method by which to optimize the transition probabilities used by the robots to traverse the structure.

[1] 2.5D is a 2D planar surface growing in a single direction of elevation

BIOGRAPHICAL SKETCH

Yiwen Hua was born in Tangshan, Hebei, China, on May 31st, 1993. At the age of 9 his family moved to Qingdao, Shandong, China. There he finished his elementary school and high school. By the time he was eighteen years old he applied Pennsylvania State University and graduated with a Bachelor of Science degree in Mechanical Engineering in 2016. Since then he has been studying at the Sibley School of Mechanical and Aerospace Engineering at Cornell University. He is expecting his Master of Science degree in May 2018.

ACKNOWLEDGEMENTS

Thanks to Kirstin H. Petersen for being my advisor, for her guidance, her wisdom, and especially for her patience in helping me find a research topic that fits my research interest and we could work on together. My thanks also go to Ross A. Knepper for serving as a member of my committee, for his invaluable feedback on the multi-robot construction in our fruitful discussions about the research. I want to give special thank to Yawen Deng for her countless contribution to my thesis as a research partner. A great deal of my gratitude is extended to the members of the Cornell Collective Embedded Intelligence Lab, for their great support and technical help at the lab and also for their friendship. I must thank Kirby Brian and Marcia Sawyer, our graduate program directors, for accepting my application to this program, answering my administrative questions with great efficiency, and providing continuous support throughout my time at Cornell. Also thanks to all my friends here in Ithaca, particularly to the graduate students from the M&AE department I have met here at Cornell, for their help in my research and their friendship that keeps me inspired through the past two years. The final thanks go to my family for providing moral and financial supports in these two years. Their existences are irreplaceable.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 TERMES and its Challenges

TERMES is an autonomous multi-robot system inspired by African mound-building termites for collective construction of user-specified 2.5D structures.[15] The algorithmic framework in the TERMES system is designed to form a construction robot swarm to build large scale structures. The TERMES robots can pick up and place custom-designed bricks, and use ”whegs”, a combination of wheels and legs, to move on unstructured terrains including lab floors, pebbles, grass, and snow. [8]. To help the robot climb and maneuver on the structure, the custom-designed bricks have notches matching the whegs and a circular indentation in the center which fits with the dimension of the robot to help it turn in place.(Figure 1.1.a) In other words, the robots and bricks have mechanical features which were co-designed to assist with the construction task. A TERMES robot has a sensing range as shown in Figure 1.1.b to obtain information about the local environment, but it does not need to, or have the ability to, communicate directly with other robots in the swarm.

The TERMES algorithmic framework consists of an offline compiler and a rule set onboard the robots. The compiler generates a map for the robots to follow based on a user-specified blueprint. The map is a connected graph of the locations on the structure, and essentially contains traffic directions that ensure that the structure grows from one point outwards. The rule set is independent of the structure, and only depends on the mechanical limitations of the robots. Together, the compiler and the rule set ensure that the structure grows in a prov-

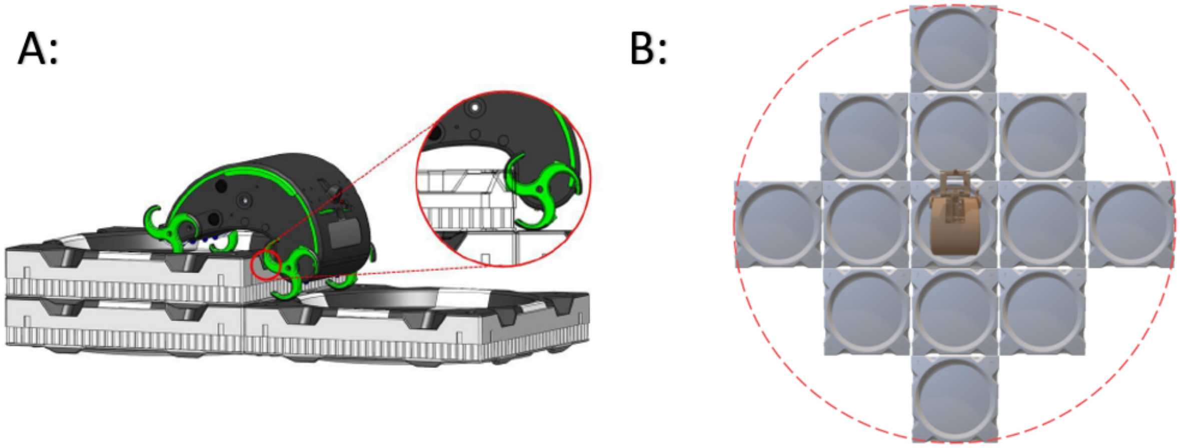

Figure 1.1: A: Illustrations of the coupled features between the robots and bricks, figure borrowed and credited to [9]. B: The sensing range of a TERMES robot when it is on the structure.

ably correct fashion, such that the final output is guaranteed to be like the user-specified blueprint. This combination of control does not require robots to be synchronized or have global knowledge of the structure. Every robot is redundant and easy to add and remove from the system.

Although the TERMES system represents a feasible solution to permit an autonomous robot swarm to construct 2.5D structures, some challenges and unsolved problems remain:

1. Due to the robots' mechanics and materials, each TERMES robot is expensive and costs about $1,800. The drive train consists of many parts, which wear quickly and are hard to maintain.
2. The compiler for generating the structure map has a computational complexity of $O(2^n)$ where n is the number of edges between sites in the structure, such that it does not scale up well with the size of the structure.

Figure 1.2: Termite-inspired robots, adapted from [15]

3. The TERMES system can assemble user-specified 2.5D structures, but it cannot handle the construction of 3D structures like overhangs, bridges, and roofs.

4. For dense structures such as a solid square, the robots make many wasted trip because they cannot find viable sites to place their bricks. This problem arises because the robots have a uniform probability of moving between neighboring bricks, and quickly transition away from the edges which must be filled in with bricks before the center of the structure.

This thesis concentrates on improving the original TERMES system in all of these aspects, and illustrates the potential of extending the TERMES system to complete large scale 3D structures.

1.2 Design Goals and Contributions

This thesis concentrates on extending the ability of the TERMES system to build 3D structures such as overhangs, and on improving the efficiency by which structures can be built, both in terms of the number of robots which can work on the structure and in terms of the size of the structures which can be built. In order to achieve this target, the following sub-goals are covered and discussed in this thesis dissertation.

The first goal is to redesign the TERMES robots to make them more affordable for prototyping and mass production. The new robot must retain the mobility of the original robot, so that it is still able to effectively traverse the structure. The new robot must also retain the ability to pick up, carry, and place the custom-designed bricks. Here, I only focus on the mechanics to reduce the cost and improve maintainability; therefore the autonomy and internal electronics are replaced by a custom-designed driver and remote control. The first contribution of this thesis, therefore, is to redesign the TERMES robot for affordability and low maintenance requirements. The mechanics of the new TERMES robot cost about \$50, compared to the original robot which is about \$1,100. The new robot also has a better maintainability, accomplished through a less complicated drive train and a motor-pulley module which can be easily replaced.

The second goal of this thesis is to extend the original TERMES system and provide additional capability to constructing 3D structures such as overhangs. The most intuitive way to solve this problem is to increase the ability (and complexity) of the robot to enable manipulation of different bricks. Here, we explore an alternative solution which is to add passive mechanical features in the bricks which allow them to expand upon deployment. This is an interesting approach because it permits the exact same minimalistic robots to build vastly more complex structures. When many robots are deployed it is especially important that each remains simple, with little wear and few breakable parts.

Besides the expanding feature, the new building material needs to retain some mechanical features from the old brick. E.g. the new expanding bricks needs to have a similar weight and size of the original bricks. Furthermore, the new bricks should keep the features which help the robots to climb, turn, and manipulate the old brick. Therefore, the second contribution of the thesis, is to design an expandable brick which the robots can use to build overhangs on existing 2.5D structures. This expandable brick is mechanically programmed such that the brick automatically unfolds when placed on top of another brick. The expandable brick inherits the necessary mechanical features from the old brick such as the circular indention at its top surface, magnets for attachment, and a handle to help the robot pick up the brick. To build overhangs on 2.5D structures with less constraints, the expandable brick should unfold and form an overhang along any normal axial direction in the horizontal plane. Moreover, for test-purposes, the expandable bricks can be manually returned to its ”non-expanded” state simply by folding it back. These design features also ensure that the robots can climb on and step off the unfolded expandable brick just like the original bricks.

The third goal is to expand on the original algorithm to permit use of these new expandable bricks to build overhangs. This new algorithm should be compiled with the old rule set, which controls the robots to build 2.5D structures, so that each robot in the swarm could use either of the control algorithms depending on the local information it has. Like the 2.5D algorithm, the new 3D algorithm needs to be linked to the onboard rule set of the robots, and ensure that no conflicts or deadlocks occur during construction. In order words, no robot can be allowed to add overhang material in such a way that it hinders the progress of future robots. Consequently, the third contribution of my thesis is to design an algorithm for 3D construction, which the robots use to build simple overhangs around convex cavities. Using the same structure map for 2.5D structures, and additional information regarding the cavity from a second off-line compiler, we show that the new framework is capable of building a pyramid-shape roof with overhangs.

The fourth goal of this thesis is to improve the transition probability in the structure map so that the robots could visit and deposit material at all the sites in the structure more efficiently. In the original TERMES system, the compiler takes the user-defined structure as input and generates a structure path for the robots to navigate the structure. The robots have a sensing range of two bricks and evaluates the local geometry to permit construction at the site it is standing on. Therefore, the probability for the robots to visit each site impacts the building speed. When a TERMES robot moves on the structure, it uniformly chooses among the next possible sites. For single-path structures, the transition probability does not affect the building speed. However, for multi-path structures the robots risk wasting many trips finding viable sites to place bricks. This is especially true for a dense structure, such as a square, where the structure has to grow from two of the four edges. Consequently, this thesis presents an optimization method to improve the transition probability in the structure map to improve the speed of construction by the robot swarm. [1]

[1]The author collaborated with Yawen Deng, a Master student at the Sibley School of Mechanical and Aerospace Engineering, on the optimization model.

CHAPTER 2

RELATED WORK

This chapter summarizes recent research efforts for automated construction, with special concentration on multi-robot systems. Generally, autonomous systems can bring many improvements and breakthroughs to the field of construction; they may replace human labor in dangerous construction environments, improve efficiency and cost of conventional construction techniques, and have the potential to build novel structures or functional structures in conditions where it is hard for humans to work. For instance, robots may rapidly construct temporary shelters or build structures like levees to reduce damage to humans and their properties in disasters situations. Up till now, however, the major usage of automation in industry is limited to human controlled gantries, cranes, and robot arms. Before automation fully enters the field, we must first prove that they are capable of long term autonomy, that they are reliable, and capable of creating novel and cost-effective buildings. Academic researchers, fortunately, get to explore beyond the scope of near-term construction, and instead focus on futuristic applications of autonomous multi-agent systems which can collaborate to address all of these goals.

The following sections present successful implementations of autonomous construction in industry and recent research work on collective construction. Research challenges of this dissertation, such as improvements and new designs on robotics hardware, custom-designed heterogeneousness building material, and construction algorithms can be found in the other chapters.

2.1 Innovative Applications of Automated Construction

Although the majority of the field of construction uses conventional methods, industrial pioneers have generated impressive accomplishments through human-robot collaborations. Broad Sustainable Building [1] completed a 57-story building in just 19 days by manually assembling prefabricated steel-frames at the construction site, demonstrating a rise of three stories per day. They also claim that their efficient manufacturing method combined with assembly and logistics can reduce the cost of a new building by 20%-40% relative to traditional techniques. For smaller-size construction, SAM [2], short for Semi-Automated-Mason, is a track-based brick laying robot for on-site masonry construction. It is designed to work with masons and improve efficiency by assisting the repetitive task of lifting and placing each brick. Masons still need to do the site setup and be responsible for the final wall quality. Beyond building walls and mortar, ICON [3], a startup company, demonstrates a more comprehensive autonomous solution, a cheap 3D-printed home, with the goal of rapidly constructing housing for people who lack shelter. Using a Vulcan printer, ICON can print an entire home for $10,000 within 47 hours. It requires a person to prepare a building base for the printer, and install windows and roofs. Besides these advanced robotics application, we can also see robots in other specialized construction tasks like tile inspections, spraying of concrete, surface finish, reinforcement, and welding. They all improve the efficiency and cost in construction while requiring human involvement and incorporation.[10, 6] In the next section, we present the research efforts that focus on automating the entire construction process with multi-agent system.

2.2 Robot Collectives in Structure Assembly

Collective construction has been an active research field for a long time. Solution are diversified into a broad range of technologies and get inspiration both from human construction and construction behavior of other creatures, such as termite colonies and families of birds. The existing collective construction systems have several benefits, such as parallelism, building structures on more than one site at once; error tolerance, the failure of one robot does not stop the process; and the possibility of building large scale and sustainable structures.[4] Therefore, the research on the collective construction grows to become an interdisciplinary field of study covering robot coordination, sensing and navigation, maneuvering and climbing, and manipulation.

The TERMES system is comprised of robots which can build structures in 2.5D by assembling custom-designed bricks [15]. In a real world demonstration, three robots completed a castle-like structure, and one robot built a structure more than 18 times its own volume. The TERMES system can build structures which are traversable by the robots, and the robots use ramps/stairs to reach higher levels. The robots use a single type of bricks to comply with the decentralized controller. Other researchers have focused on truss-climbing robots [13] capable of truss construction, fixation, and manipulation. If a centralized controller is available, the collective building team can have heterogeneous robots and various building modules. For example, a collective construction team composed of quadrotor helicopters [7] builds Special Cubic Structures(SCS) by flowing its SCS construction algorithm. Although aerial robots can build structures that are hard for ground-based robots to climb, quadrotors also typically need high accuracy global sensors like motion capture systems. As a result, it would be hard for the quadrotor team to work outside confined lab settings, especially when the quadrotors become obscured from the sensors by the growing structure.

A centralized controller has the advantage of providing accurate coordination and efficient guidance for robots to navigate complex structure, such as trusses [16, 5]. However, the agents controlled by a centralized controller need direct and frequent communication, more bandwidth is required as the number of agents increase, and eventually the efficiency margin obtained by deploying more agents will diminish. Furthermore, a centralized controller represents a single point of failure. In the contrast, the TERMES system uses distributed control schemes for assembly of bricks in 2.5D, and agents do not have direct communication. This prompts more scalability and the omission of a single point of failure, at the cost of less optimal construction.

Besides exploring types of coordination and control methodologies, another innovative research branch in autonomous construction is to give intelligence and more capability and cooperation into the building material. Some systems give changeable id's to building blocks for the robot to manipulate to guide future robots[14]; some use passive structure modules to enhance connection mechanisms[12]. In the original TERMES system[15], robots manipulate completely passive material and use co-designs among the robots, building materials, and control algorithms to improve reliability.

CHAPTER 3

ROBOTICS IMPLEMENTATION

3.1 Affordable and Low-maintenance TERMES Robot

The TERMES system is based on many homogeneous robots cooperating. To fabricate and operate many such robots, they must be affordable and easy-maintainable. The original TERMES robots cost about $1,500, $1,100 of which is due to mechanical parts because of the expensive 3D printed robot body, as well as the drive train with many steel pulleys and gears.[9] The drive train also has a high maintenance cost because these small components wear down quickly, and are hard to assemble and fit into the robot body. To solve this problem, we made a several modifications to the original robots design and the choice of materials to bring down the cost and redesign the robot to be easily maintained. First, we used cheaper material and a low-end 3D printer to print the robot body. We also replaced the metal gears and pulleys with 3D printed versions. Second, we redesigned the drive train such that the custom-designed 3D printed pulleys and motor adapters could work with the metal motor shaft and effectively transmit torques to whegs of the robots. Third, we made a special modification to the robot body so that the major components in the drive train could be fit into the robot body or be replaced with new parts through easy assembling and disassembling.

To reduce the cost of the robots, we focused on replacing the material of the robot body and redesigning the mechanics of the robot drive train. The material cost of TERMES robot was high because it had a body which is 3D printed using PolyJet Material. The mechanics cost was high because the robot drive train was composed of many off-the-shelf metal pulleys, gears, and bearings. After I conducted experiments on different robot body designs and 3D printer settings, the new robot body is printed on a low-end 3D printer using standard PLA material totaling less than \$10. Furthermore, we removed the metal components (pulleys, gears, and bearings) in robot drive train and used 3D printed ones. We printed some of them, such as motor adaptors and rear pulleys, on a 3D printer which could print parts with high precision and strength because these drive train components requires precision in assembly and strength when they transmit torques. Although the material used for these components is more expensive than the low-end 3D printing material, the unit price for each motor adapter or rear pulley is low due to their small volume. In addition, the components composing the robot arm are also printed on a low-end 3D printer. Finally, the total cost of new mechanics is about \$50, which is less than 5% of its original cost.

Instead of copying the old drive train design, we simplified it by removing all the gears and bearings and keep the necessary pulleys and motor adapters to design the new drive train. This simplification is essential for 3D printed components because the gears and bearings in the original design are too small to be printed on a low-end 3D printer. After simplification, the robot used two electric gear motors to generate power and the timing belts and pulleys to transmit power for robot motion.(Figure 3.1). These pulleys are designed to work with off-the-shelf timing belts and use an aluminum rod as an axle to connect to the whegs; the front pulleys has bolts and a counter flat surface in its axle slot to lock the relative movement between pulleys and the aluminum axle; the middle pulleys rotate around the axle that is fixed to robot body; and the back pulleys has counter features to interlock with the teeth of 3D printed motor adapters. The

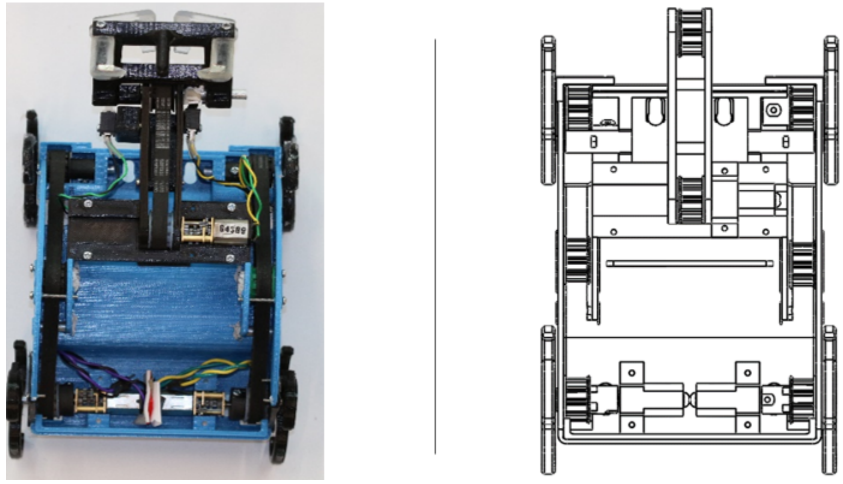

Figure 3.1: The left part of this figure shows the new drive train and 3D printing body of the robot. The right part illustrates where those pulleys and motors are positioned.

housing of pulleys and motors are printed together with the robot body. The battery pack is positioned above the electric motors to fix the electric motors in place. Furthermore, we reduce the length of the robot body because the simplified drive train is smaller than the old drive train. The new robot has a length of 138mm, and the old robot has a length of 147mm.The other dimensions of the robot body remain the same, and the whegs increase slightly to keep turning on top of bricks easy, and to make climbing easier. In the hardware testing, the new TERMES robot was successfully remote controlled to traverse the structure and manipulate both original and new bricks.

Besides the simplified and redesigned drive train, we implemented additional special designs in the new 3D printed pulleys and motor adapters shown in Figure 3.2 to improve their strength and reliability. First, we added a bolt and nut to the motor adapter to protect it when the motor shaft rotates. Through a finite element analysis and the result is shown in Fig. 3.2(b), we could see the load stress on the motor adapter is within the safe zone in the simulation. Second,

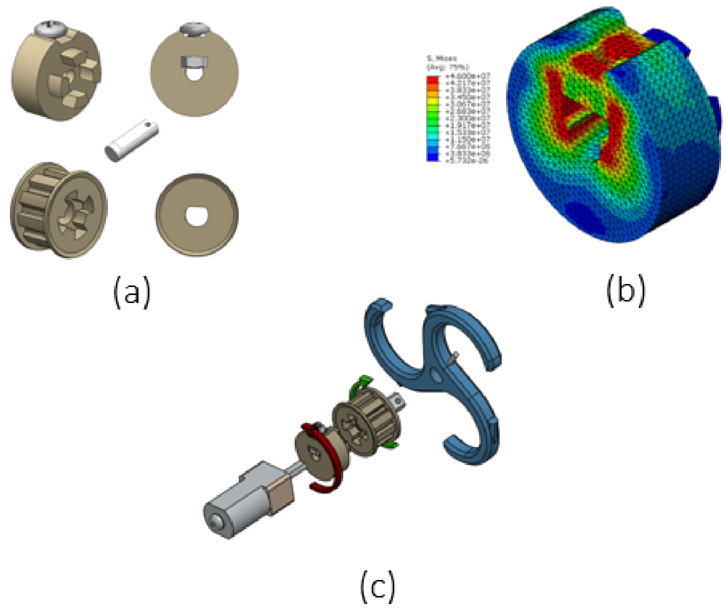

Figure 3.2: (a) shows the mechanical design of the pulley and motor adapter. We have them separated to insert the aluminum axle which connects the robot's wheel at its end. (b) is the finite analysis on the motor adapter. The rotational torque from motor shaft causes the stress concentration area. (c) demonstrates the assembly method of the rear drive train. The direction torque is shown on the assembly.

we separated the motor adapter and pulley because the aluminum axle had to be inserted into the pulley from its the counter-teeth side to lock its position. We cannot insert in through the motor adapter because its hole size is designed to fit the motor shaft, which is smaller than the aluminum axle. Third, the teeth and counter feature connection shown in Figure 3.2(a) reduces the length of the drive train assembly(Figure 3.2(c)). The shorter the assembly is, the smaller torsion stress each 3D printed part has. In the hardware experiments, the robot on average traveled 40 bricks on structure before the plastic motor adapter broke.

We also made a partial side wall of the robot's body detachable for easy maintenance. In an original TERMES robot, the motors are positioned in different heights and use gears to transmit torques to robot wheels. In contrast, the simplified drive train removed all the gears and used an electric gear motor with a higher gear ratio. In Fig. 3.1, we can see two electric motors are aligned along the back axle at the same level, and the motors are close to each other and leave little space for assembly. Considering the assembly order demonstrated in Fig. 3.2(c), we decided to make the back section of the side wall of the robot body detachable and add a "shoulder" to the detached parts such that we can use screws to attach them back. Then, the robot can be assembled with the motor-pulley module shown in Figure 3.2(c) and be installed as illustrated in Fig. 3.3. Since the motor-pulley module now can be easily disassembled, we can easily maintain the robots in a scalable multi-agent system.

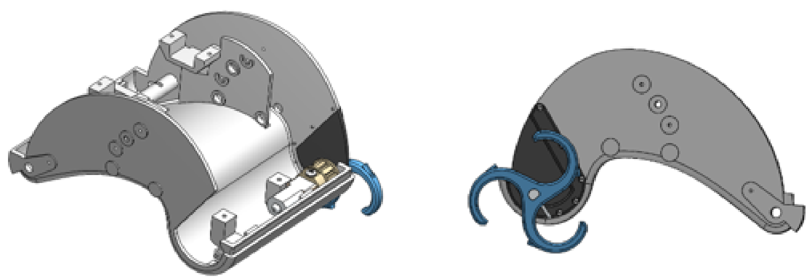

Figure 3.3: The rear drive train can be easily inserted to the robot body with the separable side wall.

By fabricating the robot with a cheaper material, simplifying the drive train, and designing an easy-assembled module, we have demonstrated the potential of using this affordable and low-maintenance robot to operate in high numbers. However, the improvement on the robot does not end here. From the design to manufacturing method, there are still numerous aspects we need to work on to improve its reliability for continuous operation. For example, we could use steel motor adapters instead of the plastic ones to increase the robustness of robot's drive train. There are only two motor adapters in each robot, therefore, we expect that the robot mechanics can remain cheap. Also, we only focused on the mechanics of the robot; another improvement would be to redesign the electronics to further lower the cost and complexity.

Figure 3.4: The photo shows the assembled, affordable, and low-maintenance robot.

3.2 Expandable Bricks

In traditional construction, many components require 3D construction including roofs, doorways, window frames, bridges, and more. The goal of this thesis is to design a new type of building material for the TERMES system, so that the new robots discussed in $Sections$ 3.1 can construct 3D structures on top of the original 2.5D structures. The TERMES system will remain decentralized for scalability and efficiency. Furthermore, we did not want to add more mechanical components to the robots, because the the added complexity would reduce the reliability and increase the maintenance cost of the robots. In this section, we would like to discuss 1) how the new building material, an expandable brick, is evolved from the old building material to remain compatible with original system; 2) the features that permit the new brick to actuate and trigger expansion; and 3) how we fabricate the expandable brick efficiently using low-cost material.

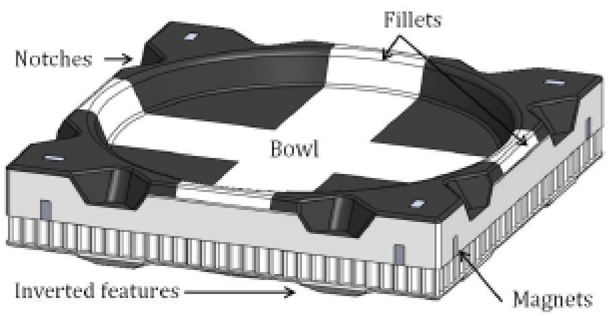

Figure 3.5: Design and Features of old brick, figure adapted from [9]. The handle is not explicitly seen from this view

In the original TERMES system, the robots use solid square bricks to build the user defined 2.5D structure. The solid square brick had necessary mechanical features shown in Fig. 3.5 to assist the robot to traverse and build the structure. The expandable brick needs to inherit some of these features and has the same engineering specifications, such as weight and dimensions of the original brick, to comply with the old brick and robots. Therefore, the body of the expandable bricks is evolved from the old brick by making a middle plane cut as shown in Figure 3.6.A and Figure 3.6.B. The old brick measures 215×215×45mm and weighs 220-240g. The new brick measures and weighs the same as the original brick, however, when unfolded it measures 215×430×22.5mm (Figure 3.6.C.

The position of the brick handle is not altered relative both to the TERMES robot and the original brick. The larger part of the handle is fastened into one half of the new brick. The original ripple pattern on the contours of the brick, was removed. This feature is to help the ultrasonic sensor of TERMES robot sense the structure when it is off the structure. Since the expandable bricks will only ever be placed on top of existing structures, these are no longer needed. Finally, the last fabrication design requirement involves the strength of the Urethane foam material that makes up the brick. The thinnest part of the half brick should be more than 5mm to avoid breaking. The fabricated half-height brick can be placed on top of an original brick with any orientation. Although the two half-height parts do not contribute additional volume per brick placed, they do form overhangs on the existing 2.5D structure once unfolded.

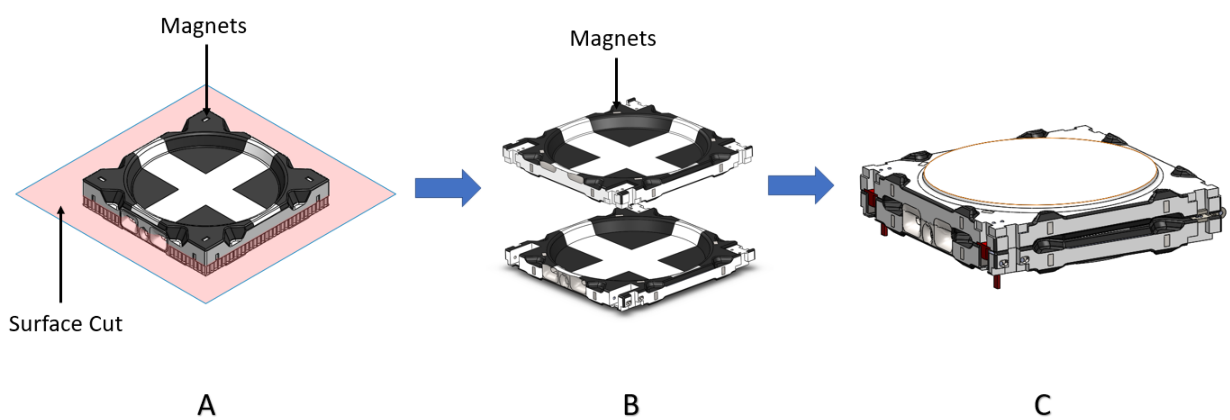

Figure 3.6: a) shows the cutting plane for separating half-height bricks and the old position of magnets. b) shows the separated half-height bricks and the new position of magnets. c) illustrates an example of how expandable bricks may be assembled.

The two half-height bricks together form the body of the expandable brick. We then design a mechanical actuator to expand the brick and a trigger mechanism to open the brick automatically when the expandable brick is placed on the top of another brick. It is not easy to design these mechanics because of the required features and the constrained engineering dimensions inherited from old bricks. After including those features and designing within the size of orig-

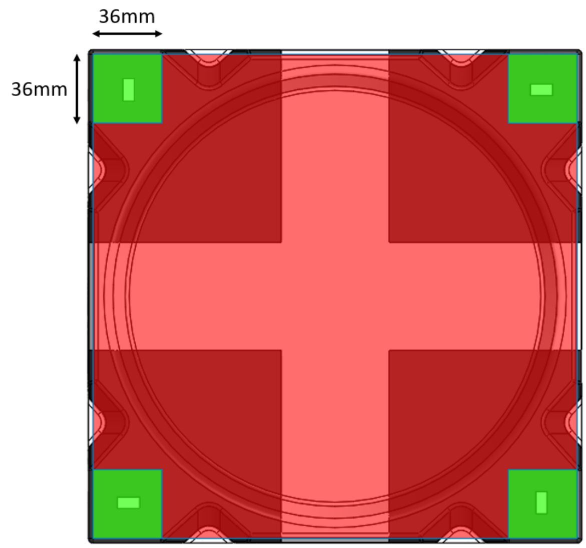

Figure 3.7: We can add additional mechanical features to the green areas. The magnets in those green areas are moved to the shoulder of the brick's circular indention

inal bricks, the design space left in the brick body are the four $30mm \times 30mm$ green patched corners as shown in Fig. 3.7. The magnets in these four corners are moved to the circular shoulder of the bowl feature on the brick as illustrated in Fig. 3.6.B.

Within the green patched design area, we designed a mechanism to provide enough torque to unfold the top half brick by assuming the bottom half is attached to the structure through magnets.

Adding power circuitry in the brick will significantly increase the fabrication cost and time; we need the expandable brick to remain cheap and easy to fabricate for mass production. Moreover, the TERMES robot has a maximum lift weight of 240g. Adding a motor and a battery pack would easily exceed this weight limit. Instead, we designed a torsion spring hinge, which is a simple and purely mechanical component, to actuate the unfolding process. The hinge bod-

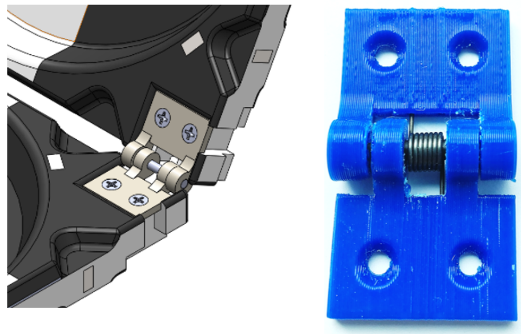

Figure 3.8: The torsion spring hinges uses screws to fix on the bricks. The hinge is printed on a low-end 3D printer. The rotational axle is an aluminum rod. The torsion spring legs fit into plastic hinges through two holes; these two holes are also 3D printed with hinge bodies.

ies are printed in a low-end 3D printer and are fixed in the design area as shown in Fig. 3.8 using screws. The torsion spring hinge plus four fixation screws and a aluminum rod weighs 9g. The hinge bodies are designed to allow the top half-height brick to rotate around the axle. When the top half-height brick rotates towards the top surface of bottom brick, it would stop until the top surfaces of these two half-height bricks meet. At this state, the expandable brick could be carried and manipulated by the robot.

Besides the hinge design, the torsion spring should provide enough torque to unfold the top half-height brick until the two half bricks lie on the same horizontal plane as demonstrated in Fig. 3.10. The torque load from top half-height brick can be easily calculated by multiplying the brick weight with the torque arm. After the expandable brick unfolds more than 90°, the torque from the brick's weight starts to apply to the same direction of unfolding. Theoretically, the torsion spring with a deflection angle of 90° and maximum torque equals to the brick's maximum load torque would unfold the expandable brick with just enough torque. This ideal torsion spring also does not provide excessive torque and slam the top half-height brick to the structure because the torsion spring starts to prevent the expandable brick from continuing to unfold once the top brick passes the opening angle of 90°. However, the 90° torsion spring often does open the expandable brick completely because of the friction between hinges and aluminum axle. Based on the torque analysis shown in *Figure* 3.9, we chose the 120° left and right-hand torsions spring made of Music-Wire Steel and that has an outside diameter of 0.315 inch, shaft diameter of 0.187 inch, wire diameter of 0.035 inch, leg length 1.250 inches, and number of coils of 5.17 to ensure the expandable brick opens completely and form a horizontal overhang.

The torsion spring hinge is a simple but robust actuator for the purpose actuating the top half-height brick. We still need a mechanism which works as a trigger to mechanical actuator. Since the expandable brick needs to remain "closed" until it is placed on the structure, the force that could be used to trigger the expansion appears when the bottom of the brick touches the top surface of another brick. The magnets for attachment and the weight of the brick itself are the major components in this triggering force. Therefore, we designed a mechanical elastic latch which can be automatically triggered when the expandable brick attaches to and becomes aligned with a brick underneath. The elastic latch design is shown in Fig. 3.11, and the size fits within the 30×30mm design space. To make the brick reusable, the latch is designed such that it can be manually reset by a person.

The mechanism in the elastic latch translates the triggering force, which is normal to the bottom surface of the brick, into a horizontal movement to open the latch (Fig. 3.11). This is achieved through a simple rotation mechanism.

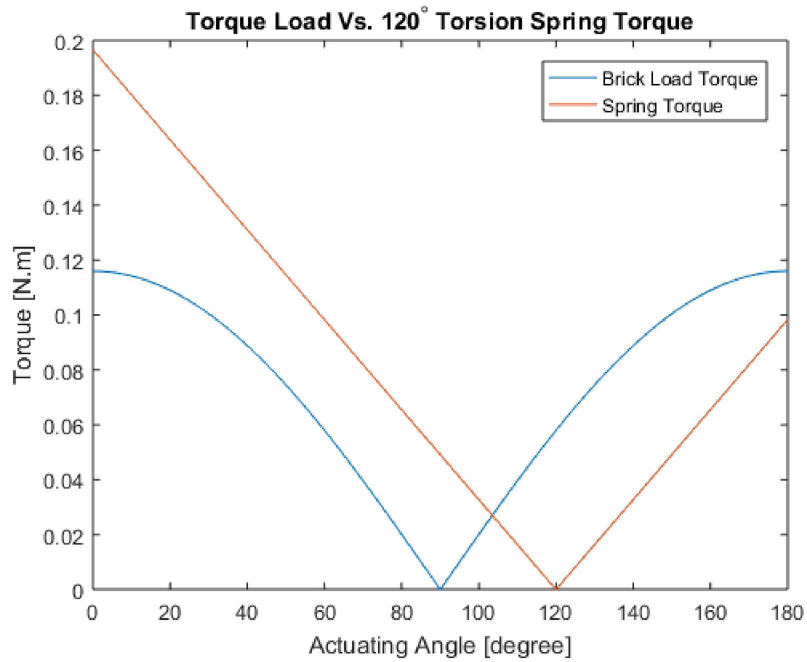

Figure 3.9: To unfold the brick, spring torque should be higher than brick load torque before the unfolding angle reaches 90°. After 90°, spring torque needs to be less than brick load torque. In the graph, it shows the torque graph of a torsion spring with a deflection angle of 120°. Since the directions of both brick torque and spring torque are reversed when they reach zero torque in the graph, the total torque acting on the unfolding direction is still position from 90° to 120°.

The rotator has an elongated arm which interacts with the trigger. The top part of this rotator connects with the slider using kite wires (not shown in the figure).The kite wire goes through the holes on the slider and the rotator. The trigger is positioned vertically and is slid into the carved slot at the bottom of the latch housing. A rubber band or a pair of compression springs push the slider to lock the latch in place. The catch is fixed by a screw to the top half of the brick. Therefore, when the trigger is pushed upward, the rotator rotates clockwise, and the slide is pulled into the housing because of the kite wire and releases the catch. This causes the two halves to open by the rotational torque

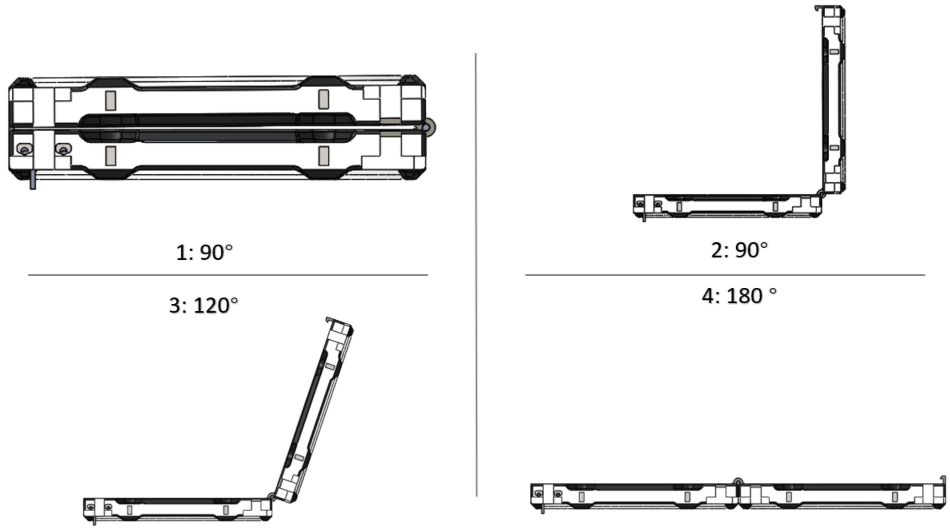

Figure 3.10: Figures 1 – 4 shows how the expandable brick unfold itself. The torsion spring finishes its actuation when the expansion reaches 90°. However, the torsion spring in the expandable brick keeps adding torque to the expandable brick until the brick reaches 120° to ensure the brick unfolds completely(180°).

from torsion spring hinges. The curved features on the slider and the catch are the designs allowing ”press lock” for minimal human effort to reset the expandable brick to its unexpanded state. Moreover, the mechanism of slider and catch reduces the risk of accidental expansion during transport.

The expandable brick is fabricated using rapid prototyping methods including casting and 3D printing. The main body of the brick is fabricated by casting a two component urethane foam which cures rigid. Top and bottom halves of the brick have different molds. The magnets are molded directly with the brick bodies and covered with ribbon to fix their positions. The ribbon provides extra friction, such that the magnets do not accidentally fall off upon attachment. The handle is further inserted into the bottom half of the brick during casting. The casting steps are shown in Fig. 3.13. Although the half bricks are fabricated sep-

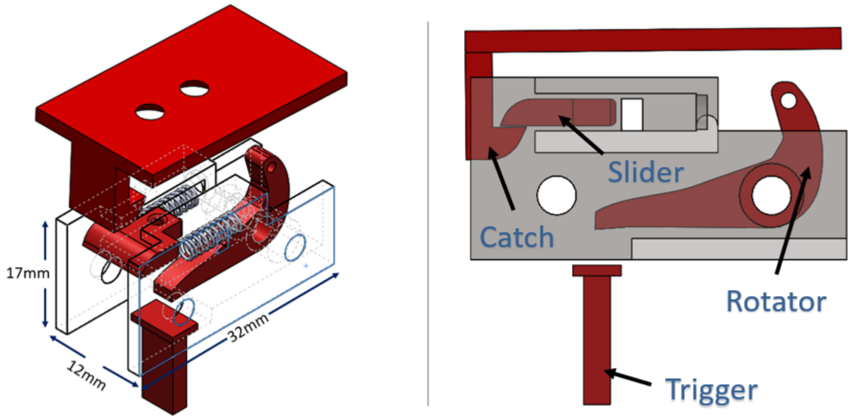

Figure 3.11: This figure shows the mechanism of the elastic latch used to trigger the unfolding of the expandable brick

arately, we can cast many of them in parallel to perform efficient fabrication. All of these new mechanical components (hinges, latches, and the handle) are designed to tolerate low precision fabrication, making them ideal for inexpensive, low-end 3D printers.

Once the foam has been cast, the torsion spring hinges and elastic latches, which are assembled using 3D printed parts, are inserted into the brick as separate functional modules for efficient production and easy maintenance. There are two steel rods inserted in the latch housing. They are longer than the width of the latch housing to restrain the movement of the latch housing relative to the foam brick. Note, one of these steel rods is also used as the rotation axle for the latch rotator. The final assembled brick is shown in 3.12. We efficiently used the limited design space to hold those mechanical components in the expandable brick.

The total cost of the expandable brick is about $25, which is the similar to the original brick. This is because we print the handle insert using low-end 3D printers to compensate the cost of additional features. The amount of urethane

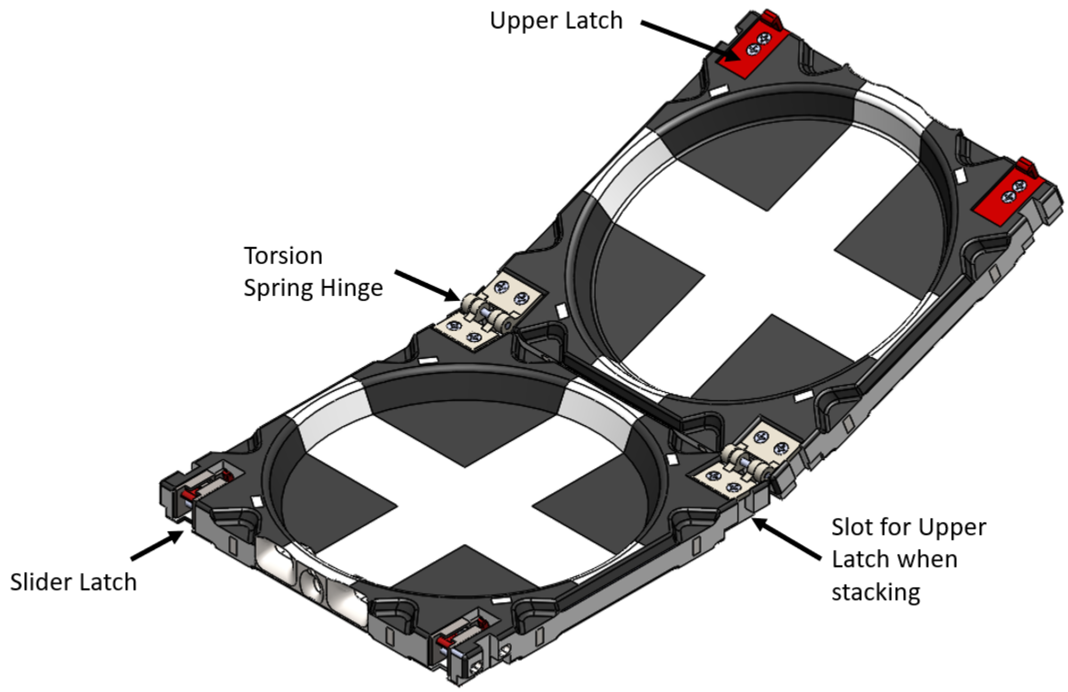

Figure 3.12: This figure shows the expandable brick assembly of half bricks, elastic latches, and torsion spring hinge.

foam required has not changed, as the total volume of the brick remains the same. The number of magnets for attachment, however, have doubled. We performed a reliability test, in which the robot was remote controlled to pick up an expandable brick, turn 90°, place the expandable brick, and let it expand automatically. We repeated this task for 10 times and have a 10/10 successful expansions for one expandable brick.

There is one more design feature to which we paid special efforts. The TERMES robot can place an expandable brick on top of an original brick in almost any orientation. The upper latch extrudes from the surface of the top brick, but these latches are designed to be positioned at the edge of the brick. Therefore, the corresponding slots carved out on the sides of the brick to fit these extruding

Figure 3.13: This figure shows the fabrication steps of casting a half brick. We add ribbon to help the magnets stay in place, and we use silicone molds with wooden frames to cast the brick out of Urethane foam.

latches upon stacking. The only position that does not match is if two expandable bricks are place immediately on top of each other in opposing directions. Since such a configuration can be replaced by two of the original bricks, we are not concerned with this deficit. In addition, we also carved out similar slots in the original bricks such that the robots can stack an old brick on an expandable brick with any orientation constraints. Therefore, the final design of expandable bricks can be used to build overhang on either the original bricks or on top of other expandable bricks.

CHAPTER 4

CONTROL ALGORITHM FOR BUILDING OVERHANGS

4.1 Original TERMES Robot Algorithm

In the original TERMES system, the robots followed a map generated by the offline compiler in order to navigate the structure. They further used an onboard rule set to determine whether building material could be added to nearby sites. This algorithm requires that 1) the robot knows its absolute position in the structure - it obtains this, by counting bricks as it moves over the structure; 2) the robot can sense information about its local environment - it does this using IR and ultrasonic transceivers; and that 3) it has access to the structure map, like the one shown in Fig. 4.4.C, which designates viable paths by which to traverse over the structure. The TERMES robot has a field of view as shown in Fig. 4.1. Yet, robots do not communicate directly to coordinate construction. This algorithm was proved to complete feasible 2.5D structures without conflicts or deadlocks [15].

4.2 Algorithmic Framework to Facilitate Overhangs

The overhang building algorithm is developed based on the framework and proofs of the original robot control algorithm, which controls the robots to build 2.5D structures using original bricks. This new algorithm allows multiple robots to build overhangs over convex cavities in an existing 2.5D structure using the expandable bricks discussed in Chapter 3. Generally, the robots would navigate

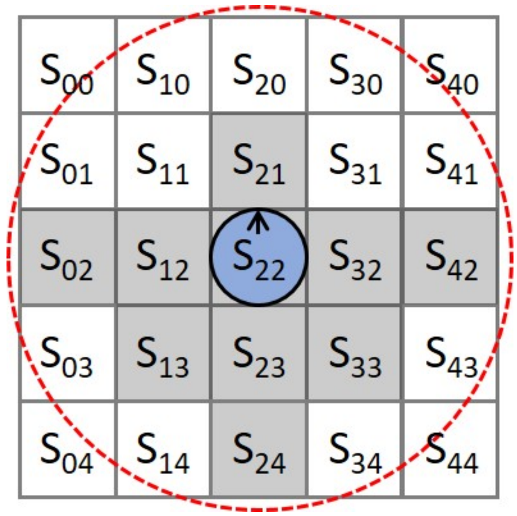

Figure 4.1: This figure shows the TERMES robot field of view.

the structure by following a similar map generated by an offline compiler, and place expandable bricks to iteratively cover the cavity. Although the structure path indicates the cavity sites are traversable, the robots could use onboard sensors to detect those cavities and does not move on them before they are covered by overhangs. The basic steps for a robot to place an expandable brick to form an overhang is demonstrated in Fig. 4.2.A-B. If a robot moves next to a cavity site, it will move to the site which is opposite to the direction of the cavity, turn around, and place an expandable brick to form the overhang to cover the cavity.

However, simply following these basic steps and placing expandable bricks immediately when the robots detect cavity sites would cause conflicts in the building process. In Figure 4.3, the robot cannot follow the map indicated by the black arrows to leave the structure. Although we can control a single robot to move against those arrows and go back to the site that has a path to leave the structure, this could cause conflicts with other robots and potential unfillable gaps in the structure. Furthermore, if robots build the overhangs without a final plan, we would likely end up with overhangs that extend overhangs from only

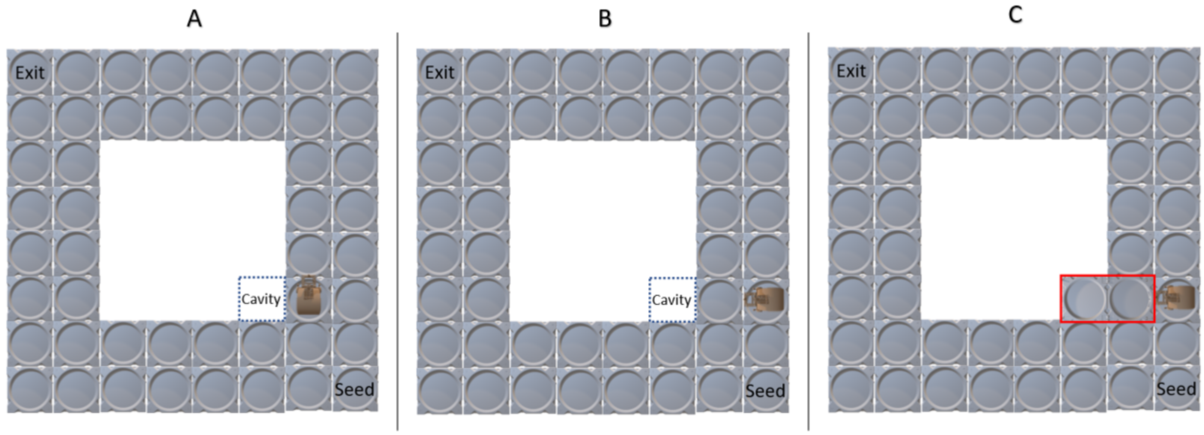

Figure 4.2: Overview of the sequence of actions to allow robots to place roof bricks. A) shows the robot detecting a cavity on its left. B) shows how the robot moving to the site which is opposite of the cavity and turning toward the cavity. C) shows the robot placing an expandable brick to cover both the site the robot evacuated and the detected cavity site.

a single direction like the situation shown in Figure 4.3. Again, robots would have difficulty leaving the structure in such cases.

To design an overhang building algorithm which does not have the problems mentioned above, we specify the robots to build overhangs layer by layer, eventually forming a pyramid-shaped roof. We noticed that the map compiled for the one-start/one-exit structure with convex cavities has the path for the robot to reach all the sites next to the cavity. Fig. 4.4.B shows an example of such a map. Robots can use this map to build the first layer of the roof. After the robots finish the first layer, the map which access the second layer is unlocked, and robots can follow the new map and continue to build the second layer, and so on.

We need extra information and constraints such that the robot collective can build the overhangs layer by layer. Robots need to know the goal structure

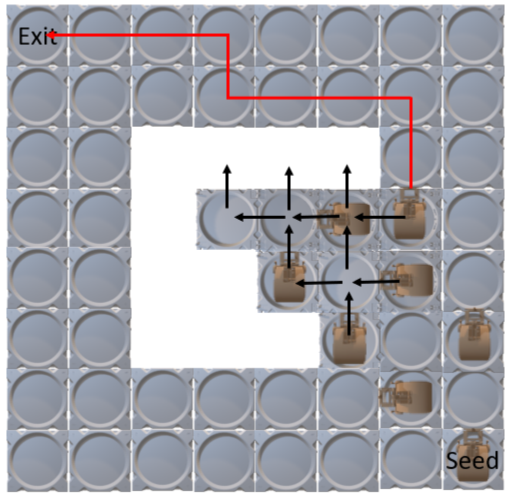

Figure 4.3: Example of how robots may become trapped and have to backtrack against the traffic directions if they are allowed to build the roof in a continuous manner. Such a path is the illustrated by the red line.

of overhangs from the compiler like the one shown in Fig. 4.4.B. To make the robot collective build each layer consecutively, the compiler also needs to label the start sites and exits sites which are the first and last cavity sites to be covered for each roof layer. Fig. 4.4.A demonstrates the start and exit sites for the roof layers which covers a 4x4 convex cavity. The roof-compiler automatically generates this information by defining the inner contour[11] of the cavity and propagating the roof start and exit sites towards the cavity center. Although the robot uses the same map as shown in Fig. 4.4.C to build both the underlying structure and the roof, only a partial map is revealed to the robot initially. When the robots start to build the overhangs over the cavity using expandable bricks, each robot remembers the current layer it is building. They then follow the partial structure path shown in Fig. 4.4.B to build the first layer of the roof. The robot will not start building the next layer until it senses that the exit-site of current layer has been built. Moreover, the robot has to check its environment to determine whether a site is valid for placing a brick. When a TERMES robot

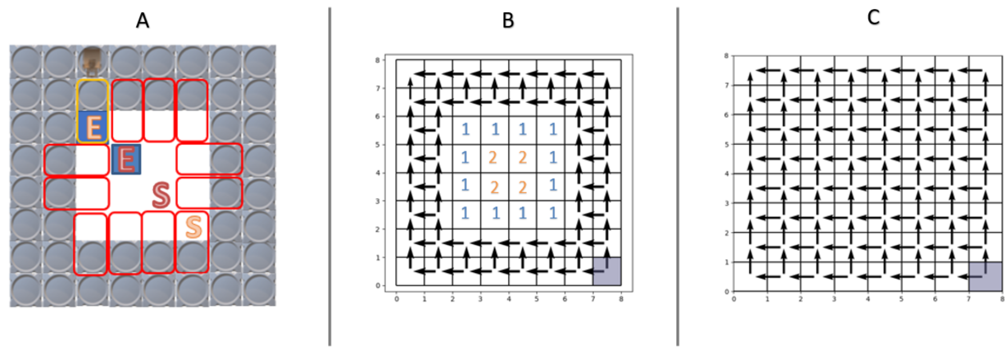

Figure 4.4: Figure.A demonstrates the roof start and exit cites propagated from the cavity corner to its center. The robot starts building the next roof layer after it has completed or observed that the current roof exit has been built. Figure.B displays the layer number of each cavity site and the partial structure path available to the robots initially. Figure.C shows a sample map compiled for a 8×8 structure. The height of this structure is one. The gray site is the start site and the top left block is the exit. For each site, the parents are those from which the arrows point in, and the children are those to which arrows point out.

detects a cavity site at either its left or right side, it will check if this cavity site belongs to the layer it is currently building to ensure overall structure is built layer by layer. The robot also checks if the cavity site is a roof start site or a neighbor of an overhang to build each roof layer continuously. The full control loop is described in A.1.

4.3 Test Structures and Limitations of the Algorithm

To test our new robot control algorithm, we simulated ten robots to build a pyramid-shaped roof which covers an 8×8 convex cavity in an existing 12×12 structure. The initial and finished structures are shown in Fig. 4.5.

For now, we only consider the one-start/one-exit structures with convex cav-

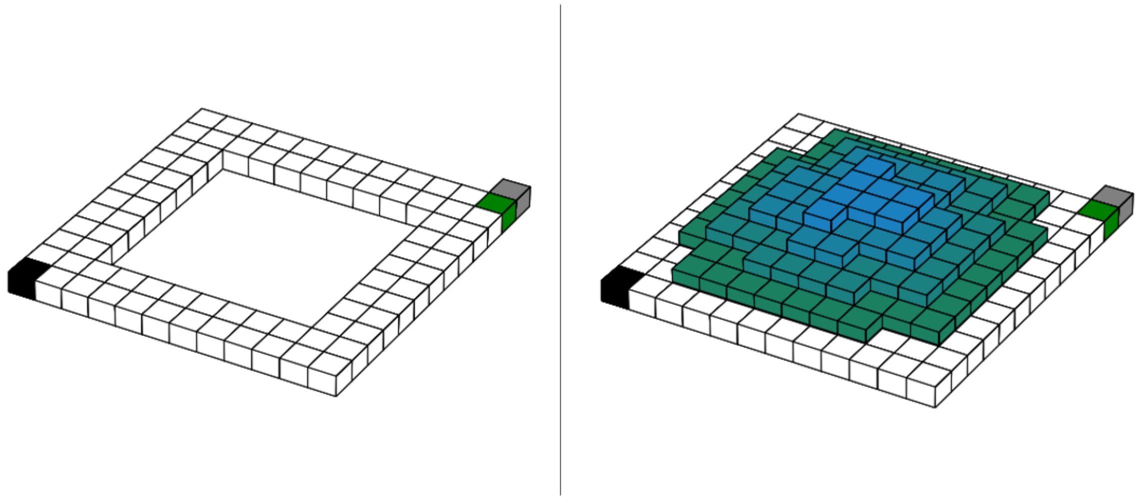

Figure 4.5: Ten robots (not shown in the figure) controlled by the overhang building algorithm use expandable bricks to construct overhangs over an 8×8 convex cavity in simulation. The green block is the start site. The black block is the exit. The gray block is the docking station for robots to get new bricks.

ities. When the compiler receives the user-defined structure, it will automatically generate the map, assign layer numbers to the cavity sites, and make sure there is a path which can navigate the robot to all the cavity sites of the current layer. As more layers are built, the robots unlock more possible sites to visit, but the requirement for having paths reaching the current cavity and leaving the structure are not changed.

Although we have yet to write up a formal proof for this algorithm, we expect this to follow the reasoning of the original proof closely. An obvious limitations of this algorithm is that it only works for convex shapes. For small concave features like Case A or B shown in Figure 4.6, the robots can follow the algorithm to build the first layer and second layer without conflicts, at which point the remaining cavity turns into a convex shape. However, if we enlarge the concave feature to Case C or Case D, the algorithm will fail because the robot cannot

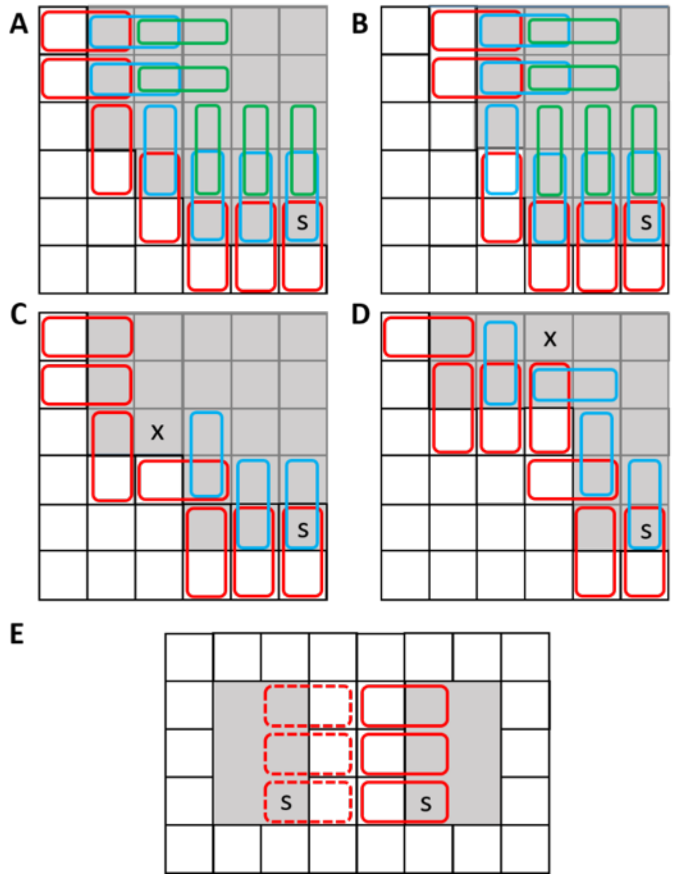

Figure 4.6: This figure shows five cases of overhang building. The blocks with darker color represent cavities. Case A and B are doable. Case C - E are not.

cover the cavities site shown in the Figure 4.6.C-D. Another limitation involves the one shown in Case E. Once a robot places an expandable brick to cover the right cavity, it cannot place expandable bricks to cover the left cavity because there is no place for it to stand when depositing the brick (Fig. 4.2.C).

This overhang building algorithm is only tested in the simulator and does not reason about the strategy of placing bricks to create structurally sound structures. The expandable bricks cannot currently be used to build large overhangs over convex cavities in practice. This is because the expandable bricks use magnets to attach to the structure, and the magnets cannot provide enough attachment force to support the weight the bricks and robots when the overhangs become larger. To solve this hardware limitation, the expandable brick could use a stronger mechanism like a mechanical lock to attach itself to the structure and other bricks, but this idea is not implemented in this thesis.

CHAPTER 5

IMPROVING TRANSITION PROBABILITY

In this chapter, I would like to discuss the transition probability, which is the probability for a robot to choose the next traversable site in a multi-path map. Firstly, I demonstrate the problem that arises if the robots select paths with uniform probability. Second, I present an optimization model to improve the transition probability to speedup the building process.

5.1 Problem for Multiple-Paths construction

In the building process, paths available to the robot are not unique because the compiler can generate multiple paths based on the specific structure. This type of map enables the robots to build the structure simultaneously at different sites. However, for a dense structure like a 15×15 solid square, the robots may make many wasted trips leaving the structure without finding a viable site to place the brick. This is because the structure has to expand from the start, and the parents sites have to be finished before the children sites can be built. We show a simulated building process for a 15×15 random height structure to illustrate this problem in Fig. 5.1.

If we take a deeper look at the reason behind the wasted trips, it is because the robots have a low probability of visiting the sites which are the parents to the rest of the unbuilt structures. Since the robots on the structure uniformly choose the next traversable child sites we can calculate the visiting probability, which is the probability for the robots to visit each site in the structure. The probability, P_i, of finding a robot in a location i, with parent locations, P_j, of finding a robot

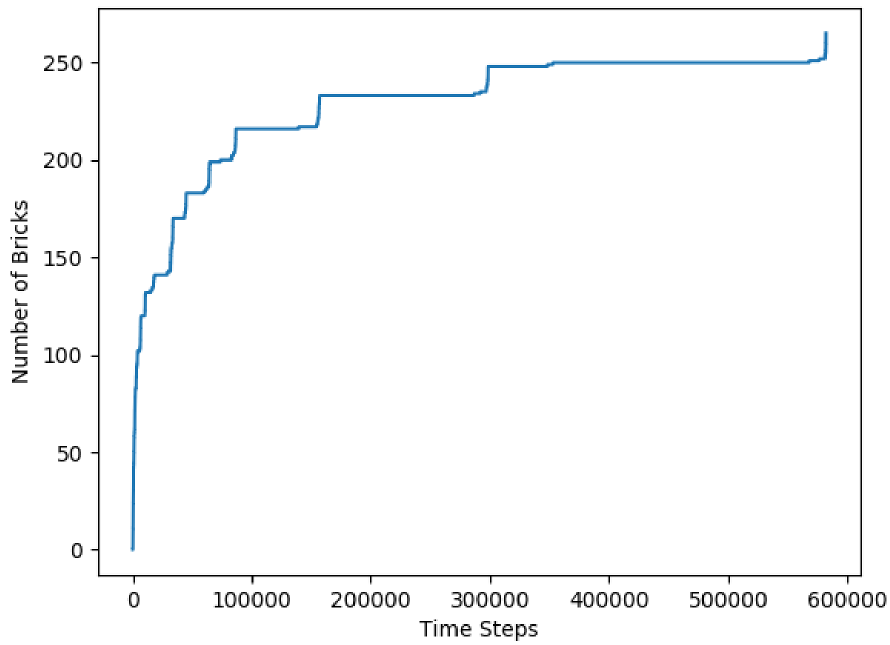

Figure 5.1: A team of 5 robots builds a dense 15x15 structure with random height. Robots make many wasted trips in the building process. In the figure, the x axis is the time step which is defined as one round in the simulator control loop, and each robot can make a move such as turn, climb, traverse, manipulate a brick, or wait. The y axis shows the number of bricks which is the cumulative number of bricks that have been added to the structure. The jumps happen when a site which has low probability for the robots to visit becomes occupied; this addition allows the robots to continue to place bricks at its children sites until another low-probability site becomes the bottleneck of construction.

in a location j, is calculated as:

$$P_i = \sum_{j=1}^{J} P_j P_{j \to i} \tag{5.1}$$

where $P_{j \to i}$ is the transition probability from site j to site i, and J denotes the total number of parents. For an one-start-one-exit structure, both P_{start} and

P_{exit} should be equal to 1. Therefore, we can recursively calculate the visiting probability from the exit. For a 15x15 square structure, the visiting probability could be as low as 0.0061%. The full visiting probability map of this sample structure can be found in the Figure 5.2.

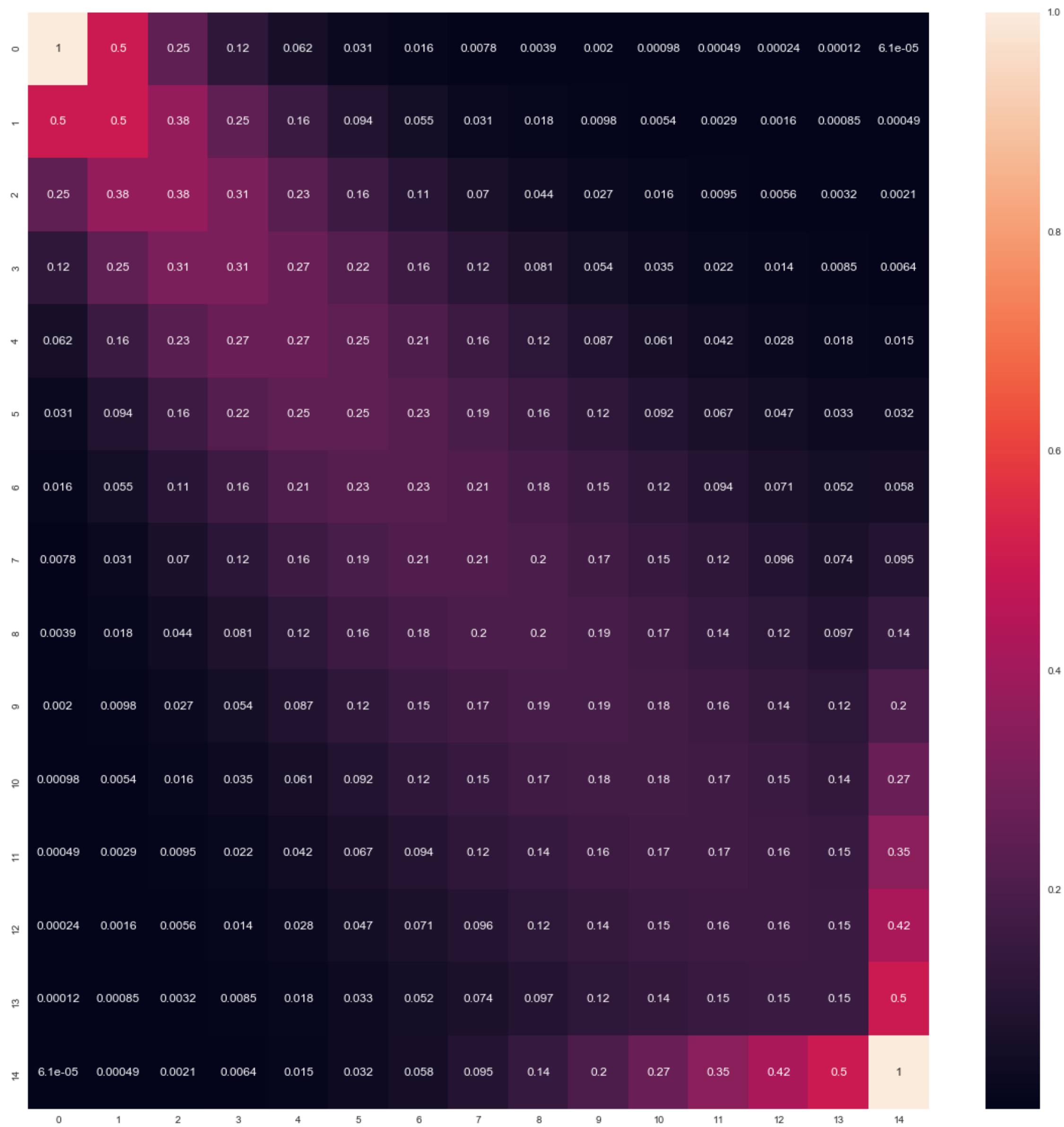

Figure 5.2: This map shows the probability of find a robot at each site if the robots on the structure have a uniform probability of choosing any children.

5.2 Transition Probability Optimization

To improve the building efficiency, we optimize the transition probability to spread the robots more efficiently over the structure. The constraints in this optimization problem are formulated as follows: 1) the probability for the robots to visit start and exit sites are "1", 2) the probability of any given traversable path must be more than 0, and 3) the sum of the outward transition probability of each site must be 1. By adding the constraints to each site in the structure based on the map, which could be interpreted as a connected graph of parents and children nodes, the constraints can be propagated from the start to the exit. The objective function is then derived by minimizing the variance, P_i, of the sites which have the same distance from the exit. This is because our goal is to avoid small visiting probabilities for each building front. For example, if an exit site has two parents sites and their visiting probabilities have a large variance, the robots would have a difficult time visiting the parent site with lower visiting probability to continue building the exit site. To enforce this objective across the structure, we traced up based on the map and minimize probability variance for parents that have the same distance to the exit. We used Sequential Least Square Programming (SLSQP) on our defined objectives and constraints to find the improved transition probability.The optimization is formulated as following:

$$\begin{aligned} \underset{P_{j\to i}}{\text{minimize}} \quad & \sum_{n=1}^{N} variance(\overline{P}_n) \\ \text{subject to} \quad & P_{start} = 1, \\ & P_i = \sum_{j=1}^{J} P_j P_{j\to i}, \\ & \sum_{k=1}^{K} P_{i\to k} = 1. \end{aligned} \tag{5.2}$$

where $\overline{P}$ denotes the set of the parents which have the same distance to the exit, n the distance in sites, $P_{i\to k}$ the transition probability from site i to site k, and K the total number of children of each site. After optimizing the transition probability, we obtain a more evenly distributed probability map shown in Fig. 5.4. In this improved probability map, the lowest visiting probability is 6.7%. Then we tested the performance of this optimized transition probability map by controlling 10 robots to build a 15x15 square structure shown in Fig. 5.3. The robots completed the structure in 581,989 steps with uniform transition probability. With the improved transition probability, the robots finished the same structure in 2,947 steps out of 10 runs indicating an over 150 times speedup.

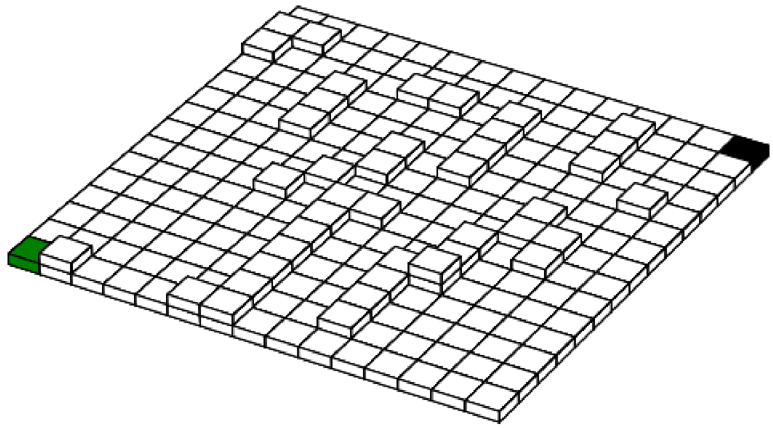

Figure 5.3: 15 × 15 structure with a total of 265 bricks, as generated from the random structure-script. In this case, the improved transition probability lowers the amount of steps taken by robots by more than 150 times.

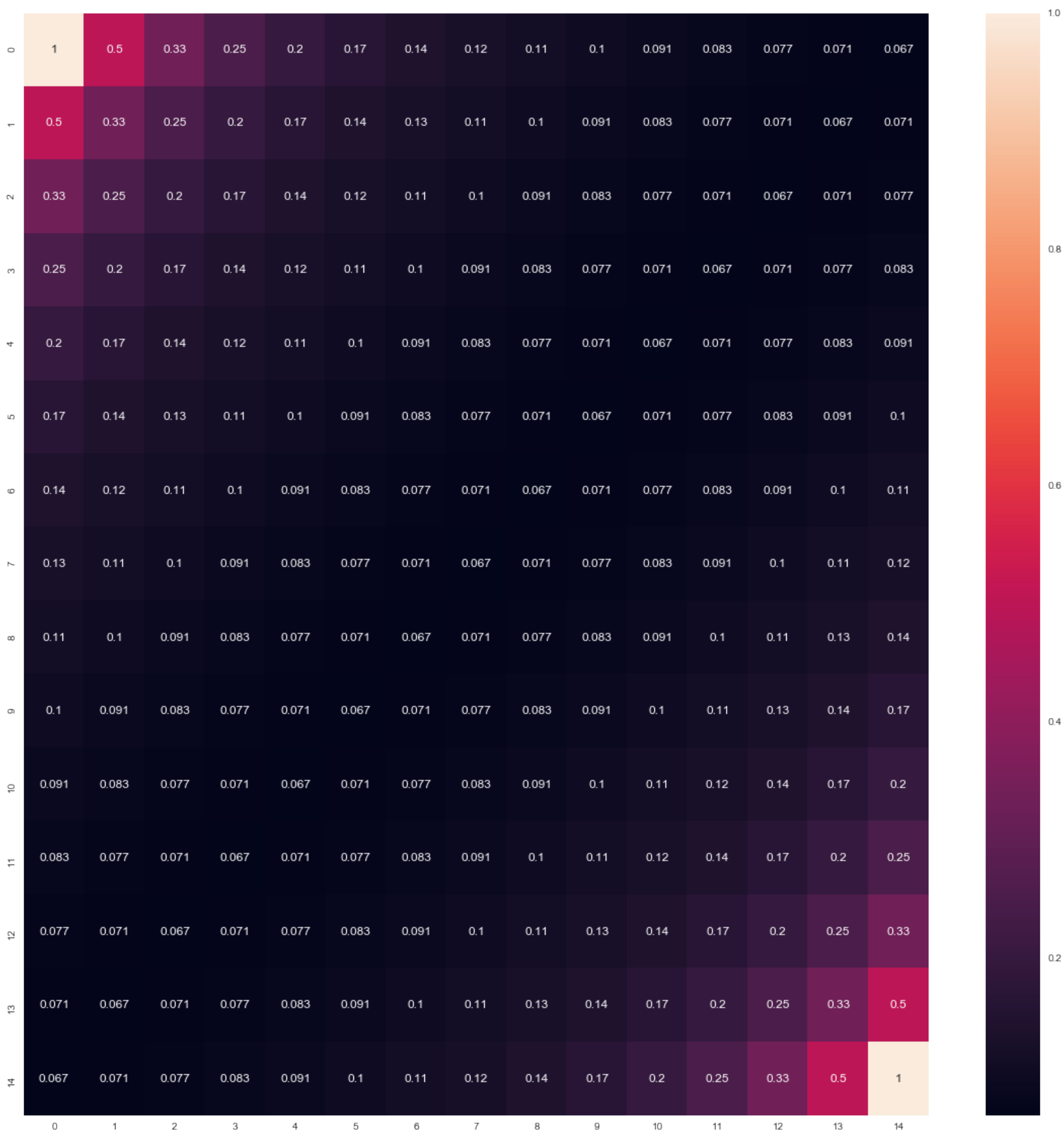

Figure 5.4: This map shows the probability of finding a robot at each site after the transition probability is optimized based on optimization model shown in equation 5.2

CHAPTER 6

CONCLUSION

6.1 Contributions

The main contribution of this thesis is improving and extending the capability of original TERMES system by 1) redesigning the hardware to yield an affordable and easy-maintenance robot, 2) designing a new building material (expandable bricks) and 3) creating the corresponding control algorithm to enable distributed construction of overhangs, and 4) improving the transition probability for construction efficiency.

The new mechanical design of the robot permits fabrication in low end, poor tolerance, 3D printers, and lowers the cost by 95%. We demonstrated that the new robots can be remotely controlled to traverse a structure of original and expandable bricks, and manipulate both of these for construction.

The new expandable bricks, can automatically unfold to form overhangs once placed upon original or new bricks. The added mechanism is achieved by a torsion spring spring hinge and an elastic latch. These added mechanical components are also manufactured using low-end 3D printers. The brick bodies can be easily fabricated through one-step casting, and the additional mechanical components can be easily assembled with the castings. The cost of a complete expandable brick is about $20, about $5 less than the original bricks.

Our addition to the robot control algorithm enables the robots to build overhangs over convex cavities layer by layer. Like the original robot control algorithm, multiple robots can build the overhangs without direct communication.

To spreading the robots over the structure more efficiently, this thesis also discussed an optimization method to improve the transition probabilities used by the robots to traverse dense structures. We showed a simple case in which this lowers the amount of steps taken by robots to complete the structure by more than 100 times.

In conclusion, we presented an improved and extended TERMES system that can build more complex structures including overhangs, bridges, and roofs by adding the heterogeneous building material and extending the building algorithm. Through this example, we have shown that the original system can be extended to a wealth of structures, while keeping to the original design philosophy of minimalistic robots which can be deployed in high numbers. It is easy to build on these minimalistic systems, to achieve even more complicated behavior - however, complexity often comes at the cost of more points of failure. By adding the passive expansion mechanism to the bricks, we expect a higher chance of success, at the expense of slightly more complicated material.

6.2 Future Work

We have replaced the complicated drive train with inexpensive 3D printed pulleys and adapters. This lowers the price of the robot, however, we have also found these pieces to be more likely to break. Further optimization is needed before these will work flawlessly. We recommend to try steel motor adapter to improve the strength of the new drive train while maintain the other inexpensive components in the drive train to keep the robot inexpensive.

The expandable bricks use magnets to help with attachment. In order to build large-scale connected overhangs like the roof in the simulation, the expandable bricks could use a new interlocking mechanisms to provide enough attachment force to support the overhangs and the robots. Based on a better interlocking mechanism, we could also take the advantage of the connection forces between expandable bricks and assemble the bricks in certain geometrical patterns to form stronger structures.

For the overhang building algorithm, it would be a great improvement if the overhangs can be built over concave cavities. If the pattern for strong overhang assembly is available, the algorithm should also encode the pattern in the building process. After making the robots autonomous and adding a new attachment feature to the expandable bricks, we would run the hardware demo again to show a roof construction like the one shown in the simulator.

Lastly, there could a further improvement on the optimization model by considering the structure height and buildable cliffs in the structure. When the robots build a randomly generated structure with untraversable cliffs as shown in Fig. 6.1, the improved transition probability only increase the building speed by 16 times. In the optimization model, we assumed all the edges are traversable, but this assumption does not hold when the robots build the cliffs which are not traversable. The optimized probability map shown in Fig. 5.4, therefore, gradually changes as the structure is expanded and untraversable edges are introduced. A possible solution to this problem could be to consider the height of each site, the probability of building the parents of the site, and the visiting probability of the site through the building process.

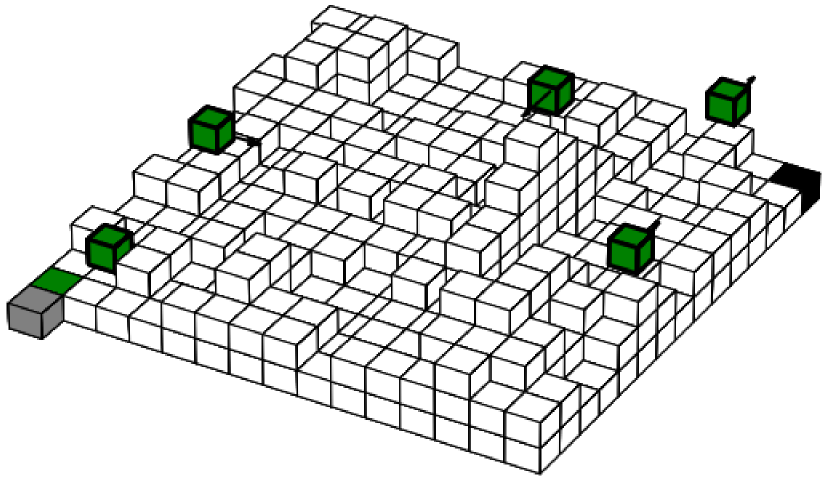

Figure 6.1: This figure shows a 15 × 15 structure with a total of 406 bricks, generated from the random script. Although this structure has the same size as the structure shown in Fig. 5.3, the improved transition probability speeds up its building process by only 16 times, which is worse than its performance with the structure shown in Fig. 5.3. This happens because of the untraversable edges in this structure.

APPENDIX A

Algorithm 1 Robot control loop for building overhangs to cover cavity of predefined structures using expandable bricks. "Parent" and "child" designations refer to the directed graph provided by the structpath; h_i is the current height of the current brick stack; H_i is the final height of the stack; the robot's current location is site 0; the cavity site is site c; The robot knows the layer height, H_L, it is building for the detected cavity site. "next" sites are children with $|H_i - H_0| \leq 1$ and $H_i \leq H_L$."start" and "end" sites are the first and last construction sites for all the layers. The key checks in line 11 are illustrated in Fig....

```
1:  loop:
2:  H_L = 1.
3:  if not holding brick then
4:      get new expandable brick from supply
5:  go to structure
6:  follow perimeter counterclockwise until entry point found
7:  climb onto structure
8:  while on structure do
9:      if the wall base is finished then
10:         move to any "next" site.
11:     if holding brick
         and either left or right of the robot, site "side", belongs to cavity.
    For the site either left or right of the robot but not the "side", give it the
    name"stand".
         and for the site "side": H_L = H_side
         and h_side ≠ H_side
         and For the site on the robot's "back" or "side back": (h_back > h_0 or
    h_sideBack > h_0 or site 0 is a "start" site)
         and site "stand" is not next to a cavity site.
         and h_0 = h_stand then
12:         move to the "stand" point
13:         place expandable brick at the site just evacuated
14:     if (robot's position is at the roof finish point where H_roof = H_roofRobot)
         and (all the "end" sites are built for the current layer robot is building)
    then
15:         H_L += 1
16:     if robot moves to one of the exits then
17:         leave the structure
```

Figure A.1: This algorithm is the robot control loop to build overhangs to cover convex cavities.

BIBLIOGRAPHY

[1] Broad sustainable building, http://en.broad.com/.

[2] Construction-robotics, http://construction-robotics.com.

[3] First permitted 3d-printed home in america, https://www.iconbuild.com/.

[4] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[5] David Hjelle and Hod Lipson. A robotically reconfigurable truss. In *Reconfigurable Mechanisms and Robots, 2009. ReMAR 2009. ASME/IFToMM International Conference on*, pages 73–78. IEEE, 2009.

[6] Changyoon Kim, Hyoungkwan Kim, Seung H Han, Changwan Kim, Moon Kyum Kim, and Sang H Park. Developing a technology roadmap for construction r&d through interdisciplinary research efforts. *Automation in Construction*, 18(3):330–337, 2009.

[7] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. Construction of cubic structures with quadrotor teams. *Proc. Robotics: Science & Systems VII*, 2011.

[8] Jeremy M Morrey, Bram Lambrecht, Andrew D Horchler, Roy E Ritzmann, and Roger D Quinn. Highly mobile and robust small quadruped robots. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 82–87. IEEE, 2003.

[9] Kirstin Hagelskjaer Petersen. *Collective construction by termite-inspired robots*. PhD thesis, Harvard University, 2014.

[10] Kamel S. Saidi, Thomas Bock, and Christos Georgoulas. *Robotics in Construction*, pages 1493–1520. Springer International Publishing, Cham, 2016.

[11] Jonghoon Seo, Seungho Chae, Jinwook Shim, Dongchul Kim, Cheolho Cheong, and Tack-Don Han. Fast contour-tracing algorithm based on a pixel-following method for image sensors. *Sensors*, 16(3):353, 2016.

[12] Yuzuru Terada and Satoshi Murata. Automatic assembly system for a large-scale modular structure-hardware design of module and assembler

robot. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2349–2355. IEEE, 2004.

[13] Marsette Vona, Carrick Detweiler, and Daniela Rus. Shady: Robust truss climbing with mechanical compliances. In *Experimental Robotics*, pages 431–440. Springer, 2008.

[14] Justin Werfel, Yaneer Bar-Yam, Daniela Rus, and Radhika Nagpal. Distributed construction by mobile robots with enhanced building blocks. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2787–2794. IEEE, 2006.

[15] Justin Werfel, Kirstin Petersen, and Radhika Nagpal. Designing collective behavior in a termite-inspired robot construction team. *Science*, 343(6172):754–758, 2014.

[16] Seung-kook Yun, David Alan Hjelle, Eric Schweikardt, Hod Lipson, and Daniela Rus. Planning the reconfiguration of grounded truss structures with truss climbing robots that carry truss elements. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1327–1333. IEEE, 2009.